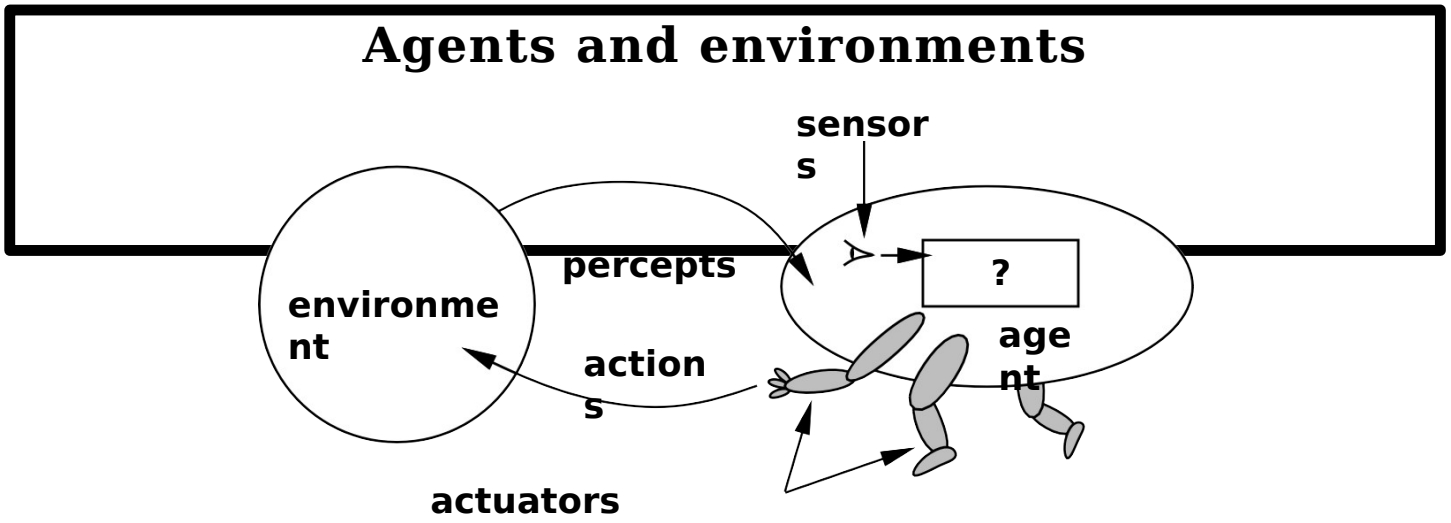
Today's class

- What's an agent?
  - Definition of an agent
  - Rationality and autonomy
  - PEAS (Performance measure, Environment, Actuators, Sensors)
  - Types of agents
  - Properties of environments

Announcements:

- Please read the assigned reading BEFORE each day's class!



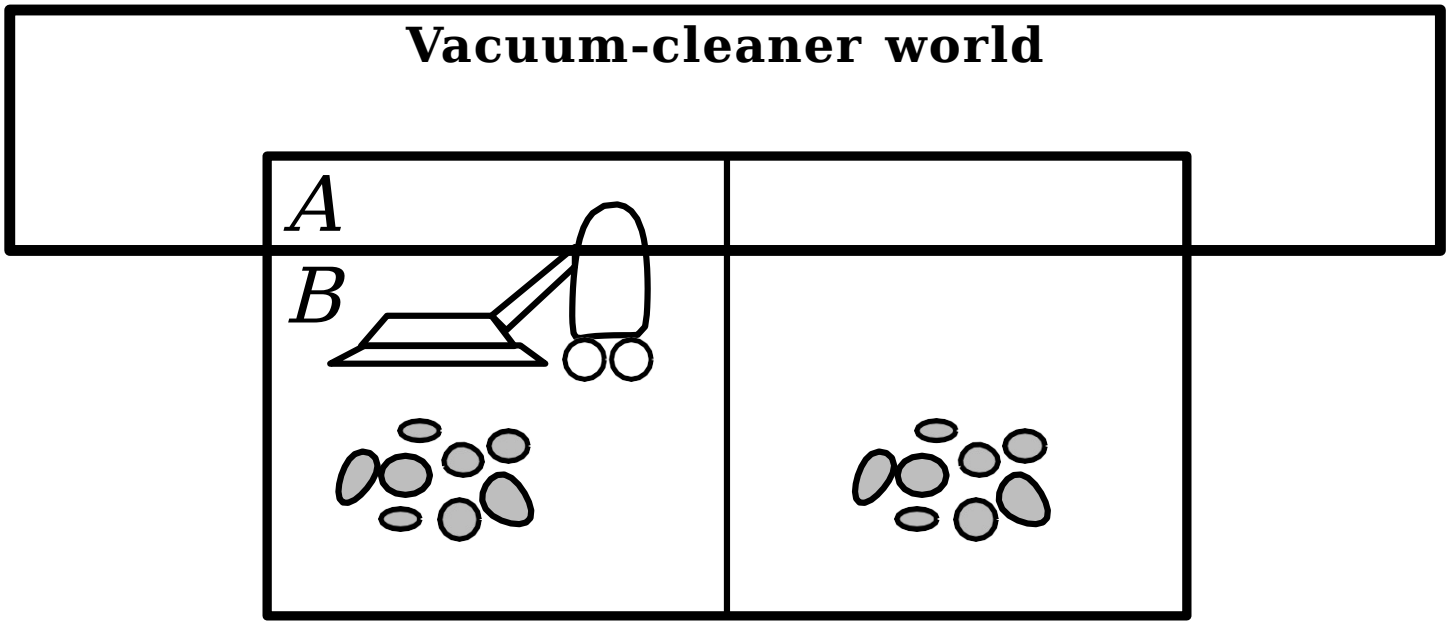
Agents include humans, robots, softbots, thermostats, etc.

An agent can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators

The agent function maps from percept histories to actions:

$$f : P^* \rightarrow A$$

Pearson



Percepts: location and contents, e.g., [*A, Dirty*]

Actions: *Left*, *Right*, *Suck*, *N oOp*

Pearson

A vacuum-cleaner agent

Percept sequence	Action
[*A, Clean*]	*Right*
[*A, Dirty*]	*Suck*
[*B, Clean*]	*Left*
[*B, Dirty*]	*Suck*
[*A, Clean*], [*A, Clean*]	*Right*
[*A, Clean*], [*A, Dirty*]	*Suck*
.	.

```
function Reflex-Vacuum-Agent( [location,status]) returns an action
    if status = Dirty then return Suck
    else if location = A then return
    Right else if location = B then
    return Left
```

What is the right function?
Can it be implemented in a small agent program?

Pearson

Rationality

What is rational at any given time depends on four things:

- The performance measure that defines the criterion of success.
- The agent's prior knowledge of the environment.
- The actions that the agent can perform.
- The agent's percept sequence to date.

Rationality concept

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rationality

Fixed performance measure evaluates the environment sequence

- one point per square cleaned up in time T?
- one point per clean square per time step, minus one per move?
- penalize for $> k$ dirty squares?

A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rational /= omniscient

- percepts may not supply all relevant information

Rational /= clairvoyant

- action outcomes may not be as expected

Hence, rational /= successful

Rational ⇒ exploration, learning, autonomy

Pearson

PEAS

To design a rational agent, we must specify the task environment

Consider, e.g., the task of designing an automated taxi: Performance measure??

Environment??

Actuators??

Sensors??

Pearson

PEAS

To design a rational agent, we must specify the task environment

Consider, e.g., the task of designing an automated taxi:

Performance measure?? Safety getting to destination, profits, legality, comfort, etc.

Environment?? US streets/freeways, traffic, pedestrians, weather, . . .

Actuators?? steering, accelerator, brake, horn, speaker/display, . . .

Sensors?? video, accelerometers, gauges, engine sensors, keyboard, GPS, . . .

Internet shopping agent

Performance measure??

Environment??

Actuators??

Sensors??

Pearson

Internet shopping agent

Performance measure?? price, quality, appropriateness, efficiency

Environment?? current and future WWW sites, vendors, shippers Actuators?? display to user, follow URL, fill in form

Sensors?? HTML pages (text, graphics, scripts)

Pearson

Other Examples

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice

Pearson

What do you mean, sensors/percepts and effectors/actions?

- Humans
  - Sensors: Eyes (vision), ears (hearing), skin (touch), tongue (gustation), nose (olfaction), neuromuscular system (proprioception)
  - Percepts:
    - At the lowest level – electrical signals from these sensors
    - After preprocessing – objects in the visual field (location, textures, colors, …), auditory streams (pitch, loudness, direction), …
  - Effectors: limbs, digits, eyes, tongue, …
  - Actions: lift a finger, turn left, walk, run, carry an object, …
- The Point: percepts and actions need to be carefully defined, possibly at different levels of abstraction

A more specific example: Automated taxi driving system

- **Percepts**:
- **Actions**:
- **Goals**:
- **Environment**:

A more specific example: Automated taxi driving system

- **Percepts**: Video, sonar, speedometer, odometer, engine sensors, keyboard input, microphone, GPS, …
- **Actions**: Steer, accelerate, brake, horn, speak/display, …
- **Goals**: Maintain safety, reach destination, maximize profits (fuel, tire wear), obey laws, provide passenger comfort, …
- **Environment**: U.S. urban streets, freeways, traffic, pedestrians, weather, customers, …

- **Different aspects of driving may require different types of**

Rationality

- An ideal **rational agent** should, for each possible percept sequence, do whatever actions will maximize its expected performance measure based on
  (1) the percept sequence, and
  (2) its built-in and acquired knowledge.
- Rationality includes information gathering, not "rational ignorance." (If you don't know something, find out!)
- Rationality → Need a performance measure to say how well a task has been achieved.
- Types of performance measures: false alarm (false positive) and false dismissal (false

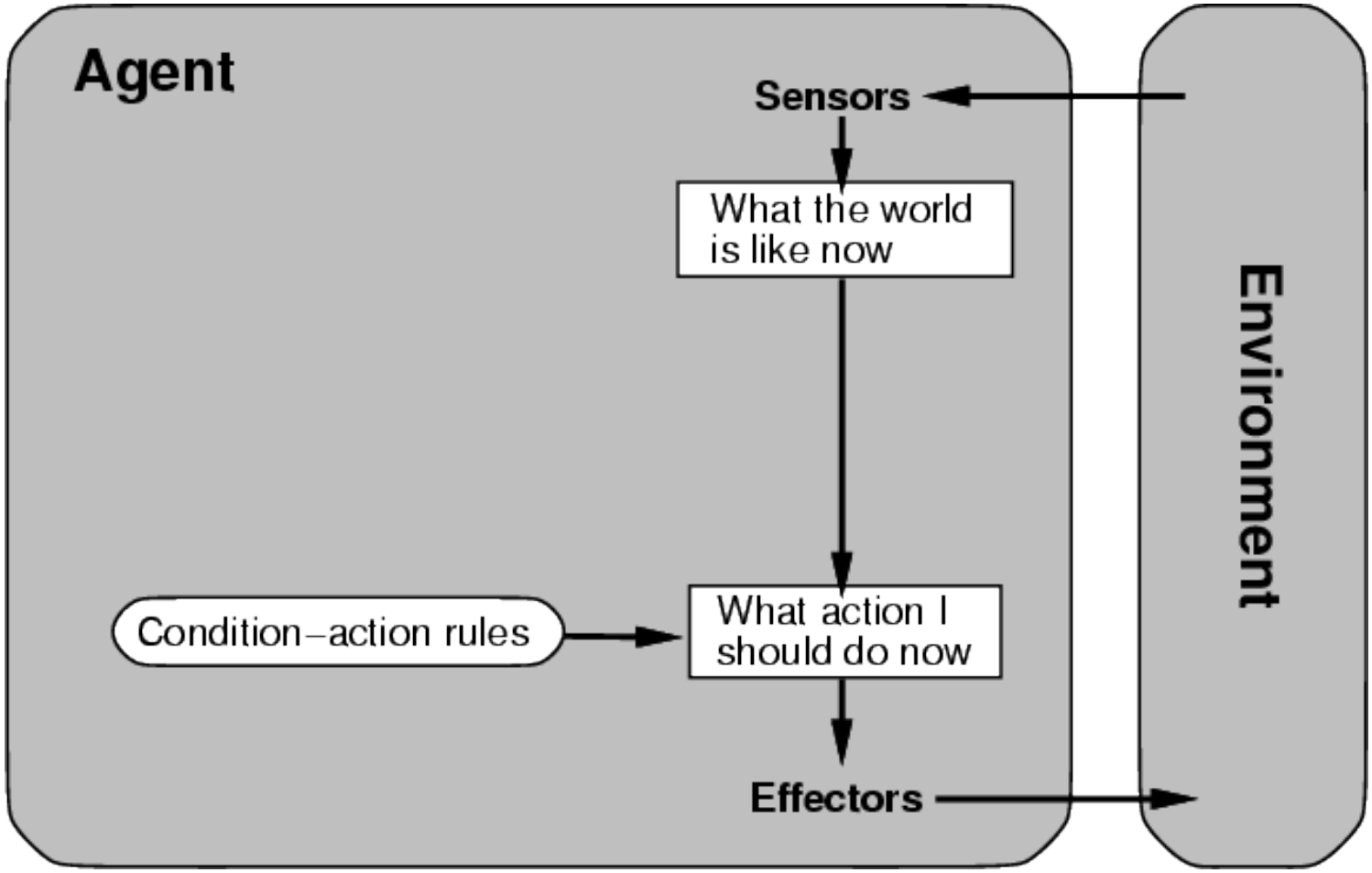
Autonomy

- A system is autonomous to the extent that its own behavior is determined by its own experience.
- Therefore, a system is not autonomous if it is guided by its designer according to a priori decisions.
- To survive, agents must have:
  - Enough built-in knowledge to survive.
  - The ability to learn.

Some agent types

- **(0) Table-driven agents**
  - use a percept sequence/action table in memory to find the next action. They are implemented by a (large) **lookup table**.
- **(1) Simple reflex agents**
  - are based on **condition-action rules**, implemented with an appropriate production system. They are stateless devices which do not have memory of past world states.
- **(2) Agents with memory**
  - have **internal state**, which is used to keep track of past states of the world.
- **(3) Agents with goals**
  - are agents that, in addition to state information, have **goal information** that describes desirable situations. Agents of this kind take future events into consideration.
- **(4) Utility-based agents**
  - base their decisions on **classic axiomatic utility**

(0/1) Table-driven/reflex agent architecture



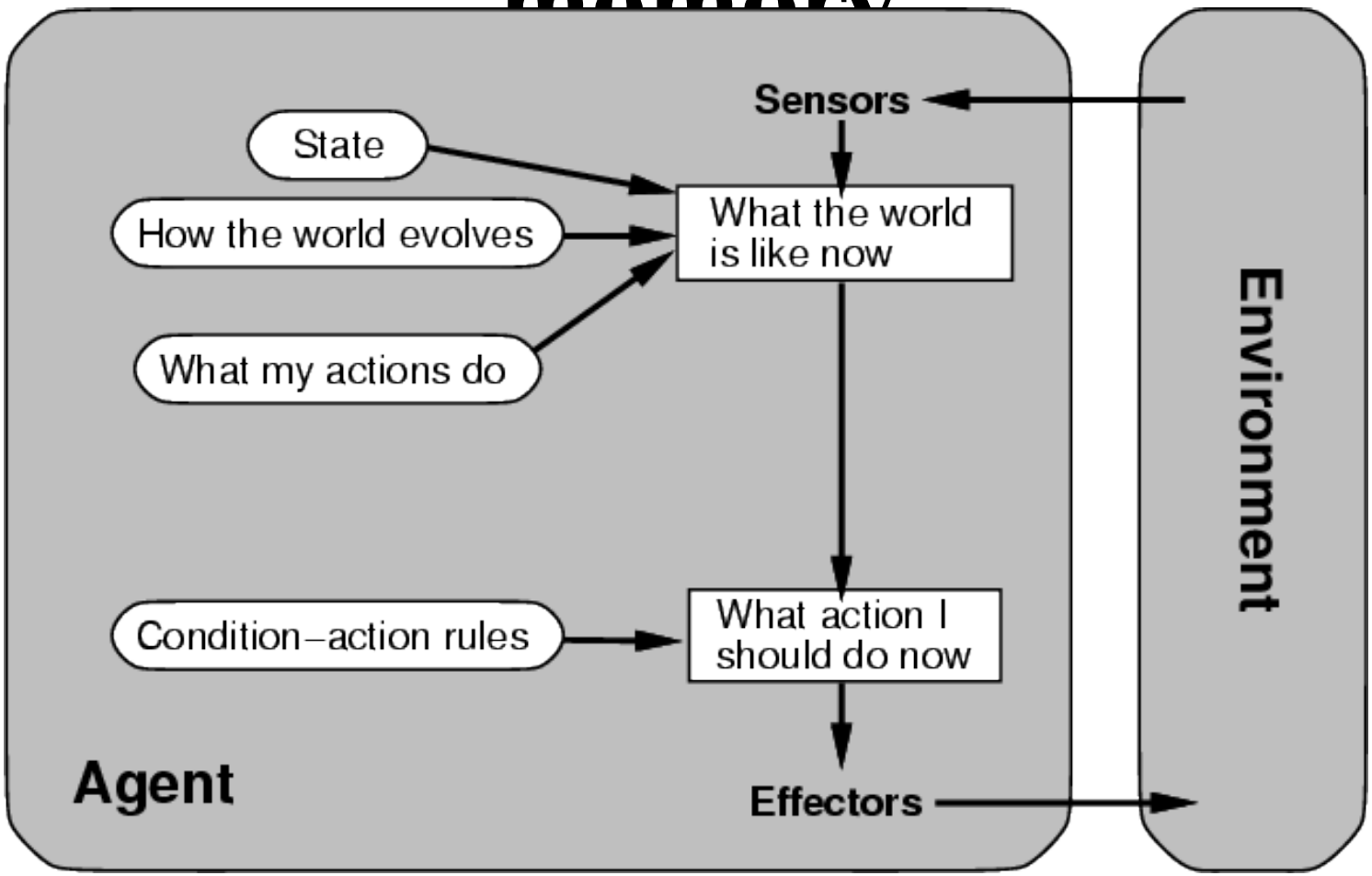
(0) Table-driven agents

- **Table lookup** of percept-action pairs mapping from every possible perceived state to the optimal action for that state
- **Problems**
  - Too big to generate and to store (Chess has about 10^{120} states, for example)
  - No knowledge of non-perceptual parts of the current state
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur
  - Looping: Can't make actions conditional on previous actions/states

(1) Simple reflex agents

- **Rule-based reasoning** to map from percepts to optimal action; each rule handles a collection of perceived states
- **Problems**
  - Still usually too big to generate and to store
  - Still no knowledge of non-perceptual parts of state
  - Still not adaptive to changes in the environment; requires collection of rules to be updated if changes occur
  - Still can’t make actions conditional on

(2) Architecture for an agent with memory



Example

function Reflex-Vacuum-Agent([*location*,*status*]) returns an action

if *status* = *Dirty* then return *Suck*
else if *location* = *A* then return *Right* else if *location* = *B* then return *Left*

```
def reflex_vacuum_agent(location, status):
    if status == "Dirty":
        return "Suck"
    elif location == "A":
        return "Right"
    else:  # location == "B"
        return "Left"

tests = [   ("A", "Dirty"), ("A", "Clean"), ("B", "Dirty"), ("B", "Clean"),
]

for location, status in tests:
    action = reflex_vacuum_agent(location, status)
    print(f"Percept=({location}, {status}) -> Action={action}")
```



```
Percept=(A, Dirty) -> Action=Suck
Percept=(A, Clean) -> Action=Right
Percept=(B, Dirty) -> Action=Suck
Percept=(B, Clean) -> Action=Left
```

Pearson

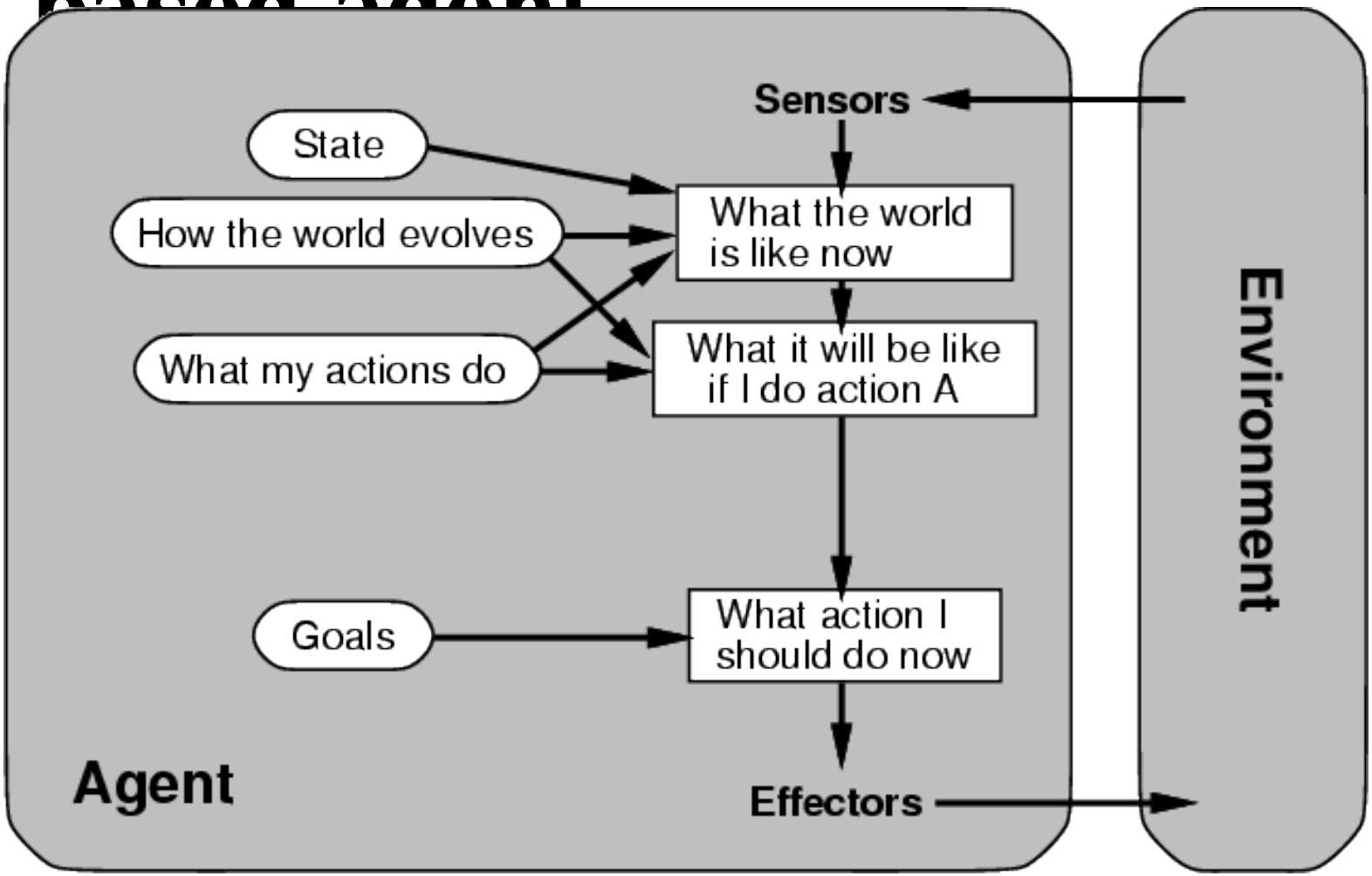
(2) Agents with memory

- Encode “internal state” of the world to remember the past as contained in earlier percepts.
- Needed because sensors do not usually give the entire state of the world at each input, so perception of the environment is captured over time. “State” is used to encode different "world states" that generate the same immediate percept.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can’t reason about hypothetical courses of action.

(2) An example: Brooks's Subsumption Architecture

- Main idea: build complex, intelligent robots by decomposing behaviors into a hierarchy of skills, each completely defining a complete percept-action cycle for one very specific task.
- Examples: avoiding contact, wandering, exploring, recognizing doorways, etc.
- Each behavior is modeled by a finite-state machine with a few states (though each state may correspond to a complex function or module).
- Behaviors are loosely coupled, asynchronous interactions.

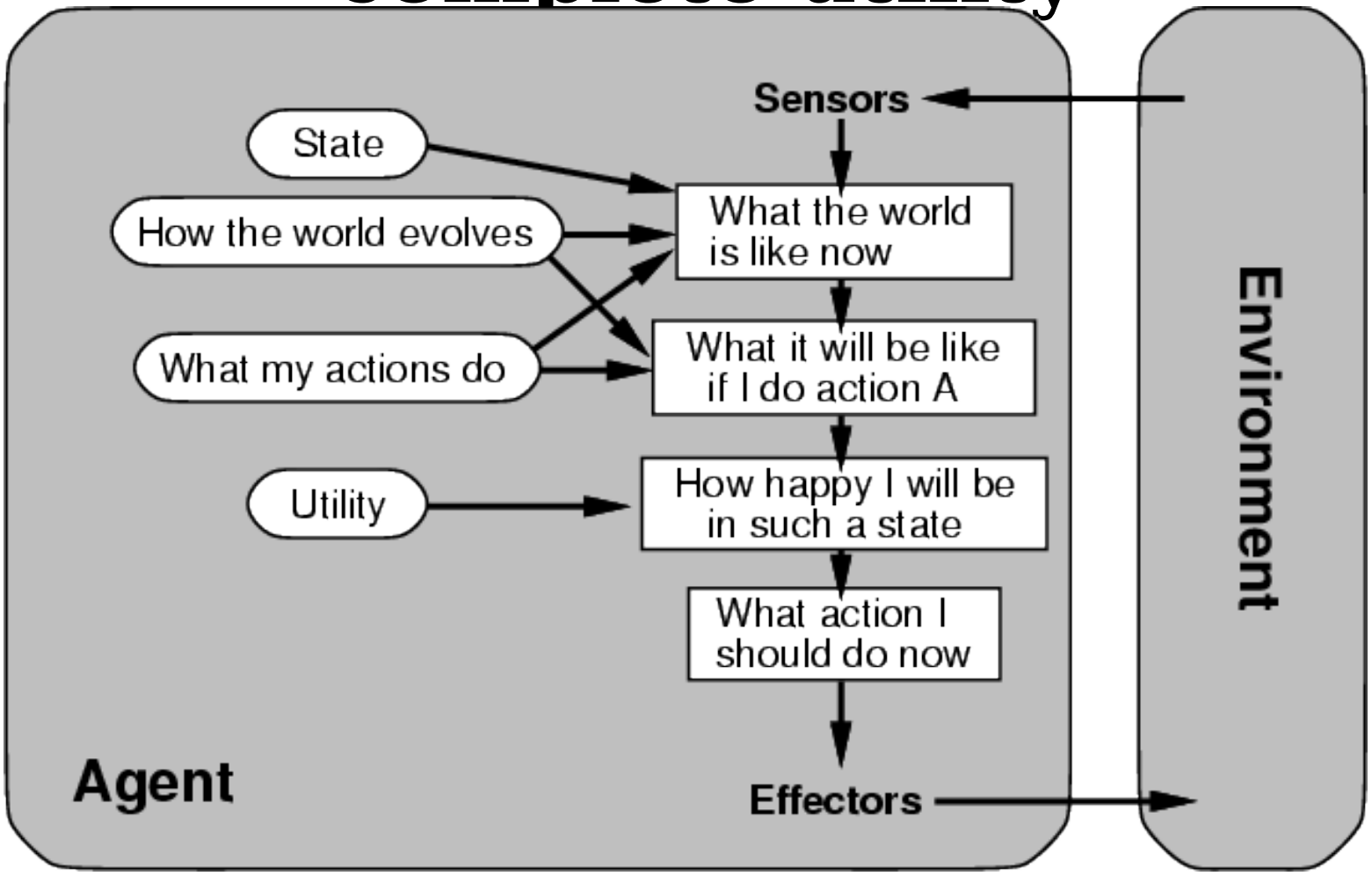
(3) Architecture for goal-based agent



(3) Goal-based agents

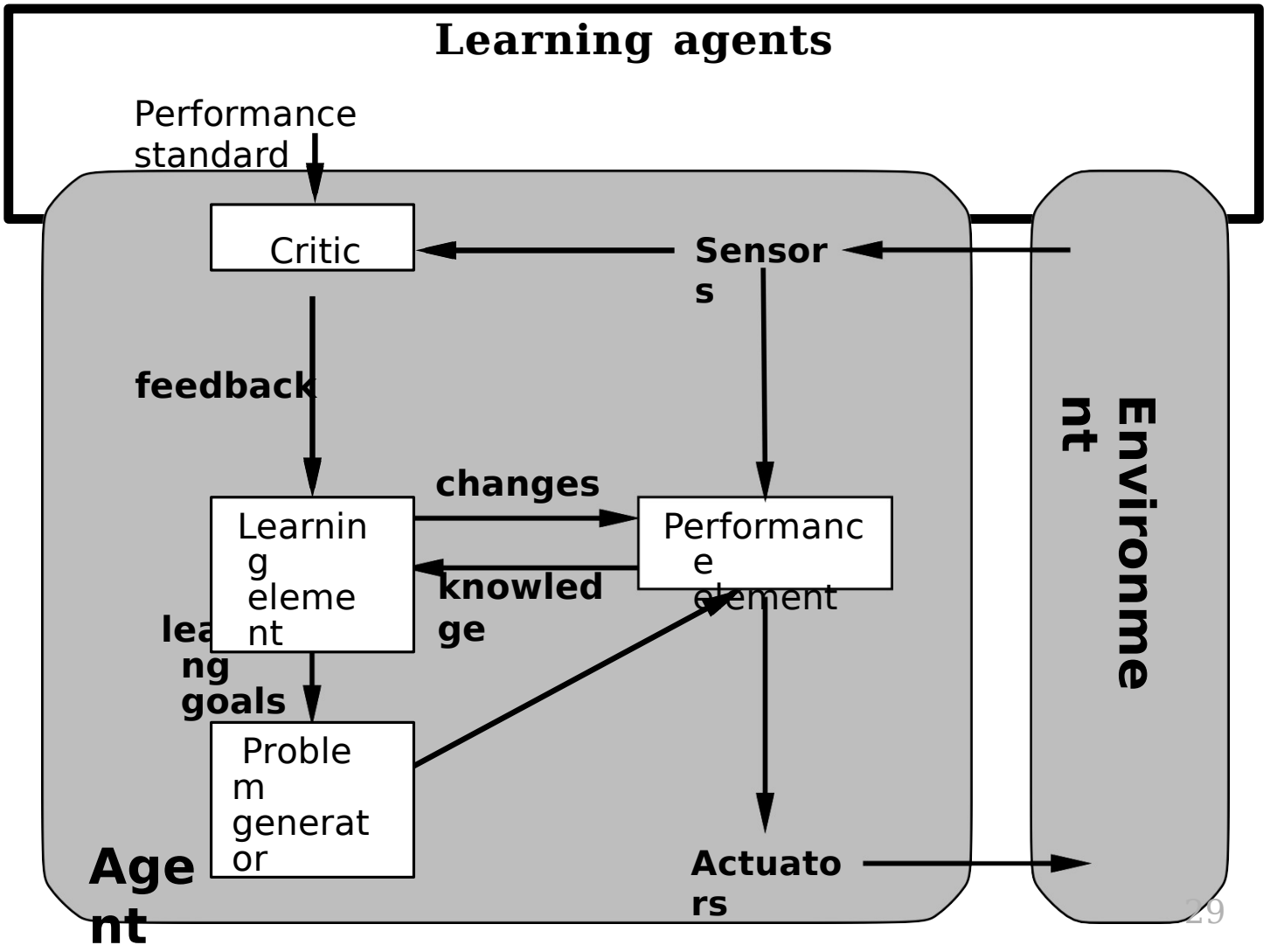
- Choose actions so as to achieve a (given or computed) goal.
- A goal is a description of a desirable situation.
- Keeping track of the current state is often not enough - need to add goals to decide which situations are good
- **Deliberative** instead of **reactive**.
- May have to consider long sequences of possible actions before deciding if goal is achieved – involves consideration of the future, *"what will happen if I do...?"*

(4) Architecture for a complete utility-



(4) Utility-based agents

- When there are multiple possible alternatives, how to decide which one is best?
- A goal specifies a crude distinction between a happy and unhappy state, but often need a more general performance measure that describes “degree of happiness.”
- Utility function **U: State → Reals** indicating a measure of success or happiness when at a given state.
- Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and


Learning agents
Performance standard
Critic
Sensors
feedback
changes
Learning element
Performance element
knowledge
learning goals
Problem generator
Agent
Actuators
Environment

Pearson

Properties of Environments

- **Fully observable/Partially observable.**
  - If an agent's sensors give it access to the complete state of the environment needed to choose an action, the environment is **fully observable**.
  - Such environments are convenient, since the agent is freed from the task of keeping track of the changes in the environment.
- **Deterministic/Stochastic**.
  - An environment is **deterministic** if the next state of the environment is completely determined by the current state of the environment and the action of the agent; in a **stochastic** environment, there are multiple, unpredictable outcomes
  - In a fully observable, deterministic environment, the agent need not deal with uncertainty.

Properties of Environments II

- **Episodic/Sequential**.
  - An **episodic** environment means that subsequent episodes do not depend on what actions occurred in previous episodes.
  - In a **sequential** environment, the agent engages in a series of connected episodes.
  - Such environments do not require the agent to plan ahead.
- **Static/Dynamic.**
  - A **static** environment does not change while the agent is thinking.
  - The passage of time as an agent deliberates is irrelevant.
  - The agent doesn't need to observe the world during deliberation.

Properties of Environments III

- **Discrete/Continuous.**
  - If the number of distinct percepts and actions is limited, the environment is **discrete**, otherwise it is **continuous**.
- **Single agent/Multi-agent.**
  - If the environment contains other intelligent agents, the agent needs to be concerned about strategic, game-theoretic aspects of the environment (for either cooperative *or* competitive agents)
  - Most engineering environments don't have multi-agent properties, whereas most social and economic systems get their complexity from the interactions of (more or less) rational agents.

Characteristics of environments

	Fully observable?	Deterministic	Episodic	Static	Discrete?	Single agent?
Solitaire						
Backgammon						
Driving						
Internet shopping						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic	Episodic	Static	Discrete?	Single agent?
Solitaire	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Backgammon						
Driving						
Internet shopping						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	**Yes**	**No**	**No**	**Yes**	**Yes**	**No**
Driving						
Internet shopping						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Driving	**No**	**No**	**No**	**No**	**No**	**No**
Internet shopping						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Driving	No	No	No	No	No	No
Internet shopping	**No**	**No**	**No**	**No**	**Yes**	**No**
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Driving	No	No	No	No	No	No
Internet shopping	No	No	No	No	Yes	No
Medical diagnosis	**No**	**No**	**No**	**No**	**No**	**Yes**

→ Lots of real-world domains fall into the hardest case!

Summary

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances The performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance Agent programs implement (some) agent functions

PEAS descriptions define task environments

Environments are categorized along several dimensions:

observable? deterministic? episodic?
static? discrete? single-agent?

Several basic agent architectures exist:

Pearson